PCT/GB 2003 / 0 0 5 5 9 8

The Patent Office

PATENTS • DESIGNS • COPYRIGHT • TRADE MARKS

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

REC'D 05 FEB 2004
WIPO PCT

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated 23 January 2004

For Official use only

THE PATENT OFFICE
A
19 DEC 2002
RECEIVED BY FAX

19DEC02 E772260-1 010092
P01/7700 0.00-0229572.3

Your reference Quality of Service (UK)

0229572.3

The Patent Office

Request for grant of a Patent

Form 1/77 Patents Act 1977

1 Title of invention

Quality of service provisioning

2. Applicant's details

[X] First or only applicant

2a If applying as a corporate body: Corporate Name

Cognima Ltd

Country

GB

2b If applying as an individual or partnership

Surname

Forenames

2c Address 131-151 Great Titchfield Street
London

827469800Z

UK Postcode W1W 5BB

Country United Kingdom

ADP Number

0055952 19 Dec 02 05:10

☐ 2d	Second applicant (if any)	
	Corporate Name	
	Country	
2e	Surname	
	Forenames	
2f	Address	
	UK Postcode	
	Country	
	ADP Number	

3 Address for service

Agent's Name	Origin Limited	
Agent's Address	52 Muswell Hill Road London	
Agent's postcode	N10 3JR	
Agent's ADP Number	C03274	727045700

0055952 19-Dec-02 05:10

4 Reference Number

Quality of Service(UK)

5 Claiming an earlier application date

An earlier filing date is claimed:

Yes ☐ No ☒

Number of earlier application or patent number

Filing date

15 (4) (Divisional) ☐ 8(3) ☐ 12(6) ☐ 37(4) ☐

6 Declaration of priority

Country of filing	Priority Application Number	Filing Date

7 Inventorship

The applicant(s) are the sole inventors/joint inventors

Yes ☐ No ☒

8 Checklist

		Continuation sheets	
Claims	0	Description	13
Abstract	0	Drawings	0

DML

Priority Documents	~~Yes~~/No
Translations of Priority Documents	~~Yes~~/No
Patents Form 7/77	~~Yes~~/No
Patents Form 9/77	~~Yes~~/No
Patents Form 10/77	~~Yes~~/No

9 Request

We request the grant of a patent on the basis of this application

Signed: Origin Limited Date: 19 December 2002

(Origin Limited)

IPR011: Quality of Service Provisioning

Rev	Date	Author	Comments
1	27th Nov 02	Ralph Greenwell	Draft for reduction and comments
2	3rd Dec 02	Ralph Greenwell	Revised following comments & updated designs

Contributors : Simon, SteveS, Mark, Andy

1. Design history

The first outlines of this design were made in the following MS-Word documents started in November 2002 by Mark Stalker and Stephen Spence:

- Priority And Timing For Object Replication
- Replication Timing Scenarios
- ID Requirements for Replication Timing and QoS
- Quality of Service Design

It should be noted that the term *QoS* is used throughout the present document, but unless stated in context this is not related to the technical meaning of *QoS* in the context of RFC2475. This IETF document defines QoS very precisely in terms of a number of metrics in the IP layer, whereas the first implementation of Cognima QoS will be applied at the application layer and will not rely upon service-specific configuration of network server parameters. For future enhancements of Cognima QoS, including interaction with IP-layer QoS, see the *Future Enhancements* section below.

2. Scheduled Replication of Data

This invention defines a way in which intelligent scheduling of data transmission across a packet-switched wireless network can be used to improve efficiency of network bandwidth usage, without seriously impairing the user experience.

Such a service enhancement needs to address conflicting requirements of various stakeholders, principally network operators, and users of Cognima enabled devices, while not placing onerous demands on the Cognima architecture.

2.1. Scheduled Replication And Network Operators

Network operators wish to smooth the peaks and troughs of the daily network usage cycle in order to make most efficient use of the bandwidth. This means moving data traffic away from peak times, and where possible moving it into troughs in the cycle. Operators will value the ability to tweak settings that affect when replication occurs, and thereby refine network efficiency.

Operators will also wish to offer services of different QoS and cost levels (expressed as a tariff) to address the varying demands of customers. This is enabled by providing the network operators with the opportunity to set dynamically, the parameters which define QoS from the users' perspective.

Most importantly the services that Operators offer should provide a compelling user experience to their customers.

2.2. Scheduled Replication And User Expectations

The Cognima technology presents new mental models for users. Replication scheduling models, and their corresponding Service plans, must be simple and consistent to aid user acceptance. Users should be shielded from the details of replication as much as possible. Data should replicate according to users' expectation.

Users want to be able to choose one QoS level from a range of options; otherwise they feel that they are either paying too much or not getting good service.

The QoS options to the user should be simple. Users will find difficulty in weighing up the relative benefits of a plan that offers *Contacts in 2 minutes, Photos in 3 hours, and Banking overnight*

against one offering *Contacts immediately, Photos overnight, and Banking in 30 minutes* etc, however much these options might fit the Network Operators' demographic research. Users will feel more comfortable choosing a general standard of service – e.g. Basic – and upgrading perhaps one particular service.

Users will appreciate the opportunity to temporarily upgrade the replication lag, by service or individual object for a cost. For example, users may want a "Send this photo now" option, which would override the default priority with which all other photographs are replicated to the Cognima server.

2.3. Scheduled Replication and Cognima Architecture

From an engineering point of view, solutions should be based on simple models rather than complex rule-sets. Rules that depend on time-outs (e.g. *each object should have its own time limit that starts counting down when it enters the Change-Log*) will severely affect performance. Likewise a solution requiring the CRE to poll the Change-Log every x seconds will also reduce performance.

3. Tariffing Options

The following two examples show how a network operator may wish to offer QoS-based tariffs to their users. The first example shows a simple tariff with little variation between data types, the second is more sophisticated and would offer greater flexibility to the user at the expense of greater complexity.

3.1. Model for simple QoS provision and Server based 'heavy-lifting'

The following model is based on simplifying the rule set whilst still offering QOS variation and appropriate user experience.

3.1.1. Model Description

There are only 3 levels of priorities

When objects are created, the parent application assigns the object one of the following priorities:

[1] Immediately

[2] When requested by server (tickled)

[3] Overnight

The Client initiates connection for Priority [1] objects

When a priority [1] object is created, the client that created it immediately opens a connection, and the object is replicated: i.e. the object doesn't wait in the change-log, unless created out of network coverage. (Phone must poll change-log for these objects every time it enters coverage)

The Server controls timing of Priority [2] and [3] objects

The Cognima Server is responsible for determining when Priority [2] and [3] objects are replicated because only it is can co-ordinate traffic from a large number of devices.

The Cognima Server tickles each Cognima device

At certain times of the day the Cognima Server tickles Cognima devices and looks for Priority [2] objects in the change-log. It replicates any Priority [2] objects that it finds.

At intervals throughout the night, the Cognima Server tickles its Cognima devices in turn, and consults its own change-log to look for Priority [3] objects. It replicates any Priority [3] objects it finds.

When a connection is open, some objects can 'queue jump'

If a connection is opened for a Priority [1] object, the Cognima Server will take the opportunity to check the change-log for other objects awaiting replication. Objects under certain size restrictions will be replicated. This is called opportunism.

The Network Operator controls QoS by varying tickling.

The Network operator can vary quality of service by increasing/decreasing the frequency of tickling for Priority [2] objects.

For example:

Basic Service:	Replicate any Priority [2] objects once per day
Standard Service:	Replicate any Priority [2] objects 2 times per day
Premium Service:	Replicate any Priority [2] objects 4 times per day
All Services:	Replicate any Priority [3] objects once a night

The user does not need to know the time of day that replication will take place, nor is replication affected by time the object entered into the change-log. However the QoS guarantees Priority [2] replication within X hours, where X < or = (number of replications contracted per day/24).

The Cognima Server may be managing thousands of devices, in which case it will carry out tickling on a rolling basis. E.g. Devices 1 to 1000 under Basic Service are tickled for Priority [3] objects at 10pm, Devices 1001 to 2000 under Basic Service are tickled at 10.15pm etc.

The Network Operator controls QoS by varying Priority assignment

The Network Operator can also vary the default QoS for all Cognima Services individually for some services, by changing the Priority value that an application assigns to the objects that it creates.

For example:

Basic Contact Service	Assign Priority [3] to new Contact objects
Standard Contact Service	Assign Priority [2] to new Contact objects
Premium Contact Service	Assign Priority [1] to new Contact objects

The priority assignment can also vary depending on the direction of data travel.

The Network Operator can influence network usage by varying Opportunism

Opportunism is taking the opportunity to replicate objects from the change-log early if a connection happens to be open. It can improve quality of service without major impact on the network. Varying opportunism will affect the frequency of tickling required, allowing the Network operator to explore the effect on network smoothing.

For example:

Replicate all objects under 750 bytes every time a connection is open

Replicate all objects under 20k every time a connection is open in off-peak hours

The user can upgrade QoS temporarily

The user should be able to force a Priority [2] or [3] object to be assigned Priority [1] status for some services, at a one-time extra cost.

3.1.2. Example scenario of a Network Operator's QoS settings

This section shows how a typical set of services might be defined for a Cognima-enabled Network Operator using the simple QoS provision.

Priority assignment

QOS level / Application	Priority assignment			
	Basic	Standard	Premium	Custom...
Messaging	[1]	[1]	[1]	
Presence	[1]	[1]	[1]	
Premium Info	[3]	[1]	[1]	
Settings	[3]	[2]	[1]	
Contacts *from* phone	[3]	[2]	[1]	

Contacts *from* PC	[3]	[3]	[2]	
Bank transactions	[3]	[2]	[1]	
Info Services	[3]	[2]	[1]	
Ringtones	[3]	[3]	[3]	
Photos	[3]	[3]	[2]	

Replication frequency

These values define how often the Cognima Server tickles devices during each day.

	Basic	Standard	Premium
Frequency	Once per day	4 times between 6am and 10pm, once overnight	8 times between 6am and 10pm, once overnight
Guarantee model	"All data is backed up within 24 hours"	"Data is backed up during daytime within 4 hours"	"Data is backed up during daytime within 2 hours"

Opportunism threshold

These values define the size threshold for clearing any objects in the change-log when a connection is open.

Object size threshold	Time of day
< 750 bytes	Between 9am and 6pm
< 20 K	Between 6am and 10pm

3.1.3. Example scenario of User Expectation and QoS

Matt's new camera-phone is Cognima-enabled. He has opted for a Basic service contract, which controls how quickly his data is backed-up. Matt knows that most of his personal data will be available on his web portal at the latest 24 hours after he enters it. He generally finds that it is available by the next morning however. He knows that he needs to leave his phone on overnight, so he has to make sure it's charged. When he goes away, he makes sure to take the lead with him. He also finds that the more he uses his phone, the quicker things seem to appear on his portal.

Matt is planning to spend the summer holidays working for his uncle's Outdoor activity centre in Cornwall. His parents want him to keep in touch and pay to upgrade his contract to the Premium service. This means that he can send them photos of what he did during the day. When he gets back from the day's walking, sailing or climbing in the evening, he accesses his web portal and the pictures he took.

3.2. Model for fully flexible QoS provision with more complex Client functionality

The following model provides more control over replication at the expense of client software simplicity.

3.2.1. Model Description

The Network Operator defines a number of replication classes

The Network Operator defines a number of Replication conditions based on:

- Time limit for replication (*e.g. [1] for immediate, [2] for within 5 minutes, [3] for overnight etc.*)
- Network usage threshold for different sizes – called Opportunism (*e.g. < 750 bytes at 80% usage, <20k at 50% usage*)
- Priority in change-log hierarchy (*e.g. Messages and Presence outrank Contacts and Settings which outrank Bank and Ringtone objects*)

The Cognima application assigns replication conditions to each object

Each object that is created is assigned the following data:

- 1 or more replication conditions
- Time of entry into change-log

The conditions may be different depending on the objects direction of travel.

Objects are replicated according to their replication conditions.

Objects sit in the change-log until one of their replication conditions is met. Objects assigned an immediate time limit (*i.e. class [1]*) will allow a connection to be initiated immediately.

The Client polls the change-log frequently looking for objects whose time limit has been reached. If it finds any, it will upgrade them to Class [1] and send immediately.

Opportunism can occur when a connection is open

If a connection is open, the client will look for any objects that meet the opportunism conditions and take advantage of the open connection to send these. This allows objects to be sent early if they are sufficiently small and the network is sufficiently quiet.

The Network Operator can offer different QoS

The Operator can set as many levels of services for the all or individual applications by defining any number of time limits, and varying the frequency of change-log polling.

The Network Operator can tweak the usage profile

The Operator can adjust the size and usage thresholds for opportunism and observe the effect on the network.

3.2.2. Example scenario of a Network Operator's QoS settings

This section shows how a typical set of services might be defined for a Cognima-enabled Network Operator using the fully flexible QoS provision

Replication conditions

These values define the main QoS service variations.

Condition	Replication
[1]	Immediately
[2]	Within 30 minutes
[3]	Within 4 hours
[4]	Within 24 hours (i.e. overnight)

Opportunism threshold

These values define the size threshold for clearing any objects in the change-log when a connection is open.

QoS	Object size threshold	Network usage threshold
Basic	< 750 bytes	70%
	< 20 k	40 %
Standard	< 750 bytes	80%
	< 20 k	50 %
Premium	< 750 bytes	100 %
	< 20 k	80%

Priority assignment

These values define the priority applied to each object that each application creates.

Object \ QoS	Conditions		
	Basic	Standard	Premium
Messaging	[1]	[1]	[1]
Presence	[1]	[1]	[1]
Premium Info	n/a	[2]	[2]
Settings	[4]	[2]	[1]

Contacts *from* phone	[4]	[3]	[2]
Contacts *from* PC	[4]	[3]	[1]
Bank transactions	[4]	[3]	[2]
Info Services	[4]	[3]	[2]
Ringtones	[4]	[4]	[4]
Photos	[4]	[4]	[4]

Polling frequency

These values define how often client polls the change-log.

	Basic	**Standard**	**Premium**
Frequency	Every 12 hours	Every 30 minutes	Every 5 minutes

4. Appendix A : Algorithmic Approach 1

4.1. Client algorithm

There are 3 priority types that can be assigned to a change log item: immediate; scheduled; overnight.

The client holds a state (a replication mode) which is set when a connection is initiated and records the priority of objects to be sent up when the connection is established.

The client change log owns 2 timers, one for scheduled connection and one for overnight connection.

The sections below describe the various triggers that are involved in QoS on the Cognima Client.

4.1.1. Change log item creation

When a new object, a property update or an object deletion are created on the device a change log item is created.

If the user has specified that the item should be immediate then the priority is set to immediate and a billing object is created (whose priority is also immediate).

Else if there is a registry setting object with a label matching the class of the object (*Rep_client_priority_class_xx*), then the priority held in that registry setting is applied to the change log item.

Else the default priority for that class is assigned to the change log item.

If the new item has immediate priority then a comms protocol connection is initiated. The mode is set to immediate. **Done.**

Else if the new item has scheduled priority then the scheduled time limit is looked up from the registry setting object with *Rep_scheduled_time_limit* label. The scheduled timer is queried. If it is active and due to expire before 'current time + scheduled time limit' then nothing more need be done. Else the timer is set to the scheduled time limit. **Done.**

Else the item has off-peak priority. If the overnight timer is already set then nothing more need be done. Else the off-peak connect time is looked up from the registry setting object with *Rep_overnight_connect_time* label. The overnight timer is set to go off at this time. **Done.**

4.1.2. Comms protocol connection established

All objects with immediate priority in the change log are sent.

If the mode is overnight then all scheduled and overnight objects in the change log are sent as well. The scheduled timer and overnight timer need to be switched off (if either is on). The mode is set back to immediate. **Done.**

Else if the mode is scheduled then all scheduled objects are sent. Updates and deletes of overnight objects are also sent. The scheduled timer needs to be switched off (if it is on). The mode is set back to immediate. **Done.** (Note: could do something more sophisticated here if really required – the timer officially only indicates that one scheduled object is due to go – we could go for some opportunism and reset the timer)

Else (mode is immediate) opportunism is used to send scheduled objects. The opportunism max size is looked up from the registry object with label *Client_rep_opportunism_size_limit*. Scheduled objects are sent until their cumulative size reaches the size limit. If there are no scheduled objects left then the scheduled timer needs to be switched off (if it was on). Otherwise it needs to be reset. Also as part of opportunism we send updates or deletes of overnight objects (which are guaranteed to be small) **Done.**

4.1.3. Scheduled timer expires

This indicates that at least one scheduled priority object is due to be sent. A comms protocol connection is initiated. The mode is set to scheduled.

4.1.4. Overnight timer expires

This indicates that at least one overnight priority object is due to be sent. A comms protocol connection is initiated. The mode is set to off-peak.

4.1.5. Tickle received

This indicates that the Cognima Server has data to send to the client. A comms protocol connection is immediately initiated. The mode is set from the tickle message.

We need to handle tickles that have been received long after they were intended, because the device was switched off or because the device was in a tunnel etc. If the tickle mode is 'overnight' but the current time is greater than a delta (*Rep_overnight_connection_delta*) away from the time in *Rep_overnight_connect_time* then we can assume the tickle was delayed.. We look up a registry object which tells us whether to connect immediately (mode set to overnight) or wait for the next overnight slot (set the overnight timer).

4.1.6. Device turned on

Timers do not survive the device being turned off and on.

As part of start up the change log is examined.

If there are any overnight objects then the overnight connection time is looked up. If we are within the overnight time delta then a connection is established, and the mode set to overnight. **Done.** Else if there are any overnight objects with timestamp before the overnight time then we can assume we have missed the daily overnight connection. We look up a registry object which tells us whether to connect immediately (mode set to overnight, **Done**) or wait for the next overnight slot (set the overnight timer). Else if there are any overnight objects we set the overnight timer.

If there are any scheduled priority objects then the scheduled timeout is looked up. If the time stamps of any objects are earlier than 'current time – scheduled timeout' then a comms protocol connection is immediately initiated, and the mode set to scheduled. **Done.** Else the scheduled timer is set to the value of 'timestamp of oldest scheduled object + scheduled timeout – current time'.

If there are any immediate priority objects then a comms protocol connection is immediately initiated, and the mode set to immediate. **Done.**

Finally if no connection has been initiated by any of the conditions above then it might be a good idea to connect anyway to allow the server to update registry objects, lock the device etc. The mode would be immediate (no opportunism) so the device wouldn't send anything.

4.1.7. Comms protocol connection failed

We need to set a reconnect timer which takes precedence over the scheduled timer. The reconnect timer can follow the current algorithm ie backing off in multiples of 2 to an upper limit.

4.1.8. Cognima Server Algorithm Description

A lot of the server side behaviour will be similar to the client. However the server has to also take a more global view to control client connection by updating the registry objects. These can be exchanged at the end of any device connection session.

The server needs an scheduled connection timer and an overnight timer for each change log.

Here are the triggers:

4.1.9. Change log item creation for a device

A new object, a property update or an object deletion is to be placed in the device change log. The change log item may have a priority assigned (eg from the CPM which created it).

Otherwise the registry object for the object's class with the label *Rep_server_priority_class_xx* is looked up for the priority (needed to allow different priorities for objects in a class going from the server to the client instead of from client to the server).

If it doesn't exist then the registry object with label *Rep_client_priority_class_xx* is looked up.

If this doesn't exist then a default value for the class is used.

If the new item has immediate priority then a tickle is sent to the client. **Done.**

Else if the new item has scheduled priority then the scheduled time limit is looked up from the registry setting object with *Rep_scheduled_time_limit* label. The scheduled timer for the device is queried. If it is active and due to expire before 'current time + scheduled time limit' then nothing more need be done. Else the scheduled timer is set to the scheduled time limit. **Done.**

Else the item has off-peak priority. If the overnight timer is already set then nothing more need be done. Else the off-peak connect time is looked up from the registry setting object with *Rep_overnight_connect_time* label. The overnight timer is set to go off at this time + 10 minutes. The idea behind this is that the tickle will only be necessary if the device itself has no off-peak objects to send. **Done.**

4.1.10. Device connects

All objects with immediate priority in the change log are sent.

If the current time is within a delta of the off-peak time then all scheduled and off-peak objects are sent. The overnight and scheduled timers are cancelled. The mode is set to normal. **Done.**

Else if the server mode is 'scheduled' all scheduled objects are sent. The scheduled timer is cancelled. The mode is set back to normal. **Done.**

Else any scheduled objects are sent, up to the opportunism size limit. If there are further scheduled objects the scheduled timer is reset. **Done.**

4.1.11. Scheduled timer expires

When the scheduled timer expires for the device an 'scheduled' tickle is sent to the device. The Server records an 'scheduled' mode.

4.1.12. Overnight timer

When the off-peak timer expires for the device an 'off-peak' tickle is sent to the device.

4.1.13. User selects or changes tariff

The registry objects for all the user's devices are created / updated. The values of *Rep_scheduled_time_limit* and *Rep_opportunism_size_limit* will get respectively decreased and increased for more expensive tariffs.

Classes can be set to one of the 3 priorities (using *Rep_client_priority_class_xx* and *Rep_server_priority_class_xx*) also depending on the tariff eg for a cheap tariff contacts may be scheduled, for a more expensive tariff they may be immediate.

4.1.14. Change in global conditions

This is the bit that isn't defined in much detail (we need the neural network :-). The Cognima Server is fed with statistics of network usage, cell loading (and anything else we can think of) and periodically (how often?) calculates the distribution of overnight connection times for all its devices. It updates the *Rep_overnight_connect_time* registry object for all devices. These get replicated to devices at the next connection.

If we wanted to we could tickle the client when the network is quiet to 'poll' for overnight changes that can go early.

4.1.15. Time specific objects

Russell mentioned the example of a traffic report that you want to arrive at 8.30, not an hour before or 10 minutes afterwards but exactly 8.30. We should control this within the CPM ie don't place the report in the changelog until you want it to go and then make it immediate.

Similarly goal flashes, half time scores etc. Make them immediate and put them in the change log when you want them to go.

5. Appendix B : Algorithmic Approach 2

5.1. Introduction

The algorithm is made up of several components.

We introduce a changelog item weight. This weight indicates how urgently a changelog item needs to be sent, the heavier the weight, the more urgent the item.

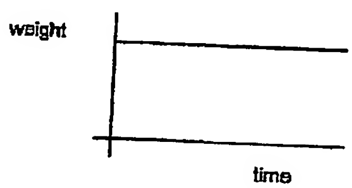
We introduce a changelog threshold. Any items that have a weight greater than the threshold need to be sent immediately.

When a connection is made all items with weight greater than the threshold minus a delta are sent. The delta represents opportunism.

For the sake of concreteness we say that weight and threshold can vary between 0 -100.

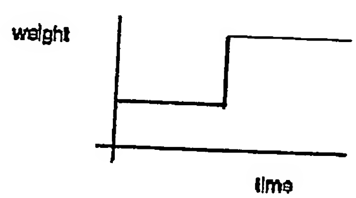
Both weight and threshold can vary over the course of a day. There will be predictable variation and also dynamic variation. Some examples will clarify this.

The weight of an item that absolutely has to be sent right now is 100.



The blue line shows that the weight is constant over time.

The weight of an item that has to go within a certain time would look like:

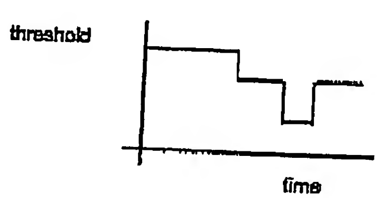


The weight of the item jumps as it reaches the time limit.

An item that should only go at a certain time will look like the diagram above but the jump comes at a particular time, not after a particular duration.

The weight of an item might change dynamically. Say for example that the device starts to run out of space: the weight of items could be increased to get them out of the changelog and allow ghosting.

The threshold will also have a graph over time:



The example above shows the threshold starting high (representing 9am – 5pm) then dropping for off peak time and dropping again for a short period which represents the time in the early hours of the morning that the device should send photos etc.

There will be dynamic changes to the threshold, for example from cell loading. If the device detected that its cell was not busy it could drop its threshold a bit which might trigger some replication.

The core of the algorithm is to calculate the threshold graph and the weight graph of every item in the change log. If the current weight of any item is greater than the current threshold then a connection is established. Otherwise the next time that any item will exceed the threshold is deduced (ie when the graphs intersect) and a timer is set for this interval.

Because both the weights and the threshold can be dynamic there are several events that can trigger a recalculation:

- A new change log item is added
- A timer expires
- The cell loading changes
- The available memory falls below a certain level
- Connection terminated

So there are two separate calculations: the weight graph of an item and the threshold graph.

Parameters that can affect the weight graph of an item are:

- Direction (client -> server or server -> client)
- Shelf life (encoded in the class?)
- Overwritable (encoded in the class?)
- Size
- Time entered change log
- Priority
- Time out interval
- Assigned time for replication
- Memory available.

Parameters that can affect the threshold graph of a change log are:

- Time of day
- Roaming
- Cell loading
- Time since last replication
- User tariff

We also need a calculation to find the time interval to the next intersection between a weight graph and a threshold graph.

5.2. Banding

We control both the client weight graph and the threshold graph by a structure we call **banding**. We assume that all graphs can be described as sets of horizontal lines which jump (concatenations of step functions). Therefore any graph can be described by an array of *bands*. A band is parameterised as follows:

Parameter	Values	Description
Band type	Delta; absolute	Whether the band is measured from creation time or against system time.

Direction	Server->client; client->server	Direction of travel of item.
Start time	Time	If delta type a duration, if absolute then a clock time.
End time	Time	If delta type a duration, if absolute then a clock time.
Weight	0 – 100	Item urgency.

Bands can be defined as deltas from a start time or against the system clock (absolute).

A weight graph for a class which should be scheduled to go within 2 hours of object creation could be described by a pair of bands:

Band1	Delta; client->server; 00:00; 02:00; 25
Band2	Delta; client->server; 02:00; 24:00; 90

A threshold graph which describes peak time from 9am to 5pm, off peak time otherwise and desired replication time as 2:30am - 3am would be described by 4 bands:

Band1	Absolute; client->server; 17:00; 02:30; 60
Band2	Absolute; client->server; 02:30; 03:00; 10
Band3	Absolute; client->server; 03:00; 09:00; 60
Band4	Absolute; client->server; 09:00; 17:00; 90

We can create QoS objects for users. Each object holds an array of bands to describe a single weight graph. A user's device might hold a QoS object for each class plus a threshold object.

5.3. Changelog item weight calculation

The weight of an item in the change log has to be calculated. The QoS object for the item's class is retrieved. The banding structure is examined and a weight is looked up from the weight graph (either by calculating current time – created time in the case of delta band type, or by comparing system time in the case of absolute band type).

To include object size as a parameter we could add two more fields to the banding structure.

Parameter	Values	Description
Size limit	Size in bytes	Object size limit over which the oversize weight applies.
Oversize weight	0 – 100	Weight of oversize objects.

If the object size is greater than the size limit then the oversize weight is used.

We could also apply a weight change depending on the memory available.

The object weight can be overridden by a user request (effectively a 'send now' button) which sets the weight to 100.

5.4. Changelog threshold calculation

The current changelog threshold weight can be extracted from the threshold QoS object using the current system time. This value can then be modified by dynamic variables. If the device can detect roaming status then this can influence the threshold (presumably finding yourself abroad should increase the threshold). If the cell loading of the local cell is below some value (eg 70%) then the threshold can be reduced. If the cell loading is above some value (eg 95%) then the threshold can be increased.

5.5. Next connection time calculation

The weight graph of an item needs to be compared to the change log threshold graph to find the next time that the item weight >= threshold weight. This calculation will ignore dynamic change to the threshold weight which is by definition unpredictable. It should be a fairly simple calculation to match up the bands and compare weights at consistent times.